NAME

MSl12 − calibrate and atmospherically correct ocean color data

SYNOPSIS

MSl12 par=file

− or −

MSl12 ifile=ifile ofile1=ofile1

− or −

MSl12 ifile=ifile ofile1=ofile1 [def_l2prod_file=def_l2prod_file] [l2prod1=l2prod1] [ofile[#]=ofile[#]] [l2prod[#]=l2prod[#]] [spixl=spixl] [epixl=epixl] [dpixl=dpixl] [sline=sline] [eline=eline] [dline=dline] [ctl_pt_incr=ctl_pt_incr] [aer_opt=aer_opt] [aer_iter_min=aer_iter_min] [aer_iter_max=aer_iter_max] [tau_a=tau_a] [glint_opt=glint_opt] [outband_opt=outband_opt] [oxaband_opt=oxaband_opt] [carder_opt=carder_opt] [filter_opt=filter_opt] [filter_file=filter_file] [calfile=calfile] [vcal_opt=vcal_opt] [offset=offset] [gain=gain] [albedo=albedo] [glint_thresh=glint_thresh] [absaer=absaer] [sunzen=sunzen] [satzen=satzen] [epsmin=epsmin] [epsmax=epsmax] [tauamax=tauamax] [nlwmin=nlwmin] [wsmax=wsmax] [maskland=maskland] [maskbath=maskbath] [maskcloud=maskcloud] [maskglint=maskglint] [masksunzen=masksunzen] [masksatzen=masksatzen] [maskhilt=maskhilt] [maskstlight=maskstlight] [sl_frac=sl_frac] [sl_pixl=sl_pixl] [met1=met1] [met2=met2] [met3=met3] [ozone1=ozone1] [ozone2=ozone2] [ozone3=ozone3] [land=land] [water=water] [elev=elev]

DESCRIPTION

This program is capable of performing atmospheric correction of top-of-atmosphere (TOA) radiances from several ocean remote sensing, spaceborne spectrometers, including SeaWiFS and MOS, and deriving atmospheric and bio-optical properities using identical algorithms for both sensors. Data input format and sensor identification are automatically determined by the program, which presently recognizes SeaWiFS Level−1A or Level−1B, and MOS Level−1B. Sensor dependent details such as band-pass-weighted quantities are included in a sensor-specific external data file, and pre-computed sensor-specific look-up tables are provided for Rayleigh scattering and Rayleigh-aerosol transmittance. Aerosol model tables make use of SeaWiFS-specific coefficients, with some adjustment of the model epsilons to correct for deviations from SeaWiFS center wavelengths. The use of pre-computed SeaWiFS aerosol tables limits the ability of MSl12 to perform atmospheric correction to sensors that do not signficantly deviate from SeaWiFS wavelengths, or sensors that contain no more than six wavelengths in the 400−700 nm.

PRODUCTS

This table contains all the products (don't get dizzy) which are available from this one program! Most outputs are 2-D arrays stored in an HDF file as a Scientific Data Sets (SDS) with the given name. The products which contain *nnn* are available at each wavelength of the given sensor. For SeaWiFs these are: 412, 443, 490, 510, 555, 670, 765, and 865; for MOS these are: 408, 443, 485, 520, 570, 685, 750, and 870; and for OCTS these are: 412, 443, 490, 520, 565, 670, 765, and 865. For the eps_*nnn_lll* product, *lll* represents the longest wavength, that is, 865 for SeaWiFS and OCTS or 870 for MOS. Note that some products have multiple names for the same exact product. This is due to the nature of HDF's SDS interface wherein each SDS must have a unique name. Duplicate names are used for compatability between several systems which use the same code (SeaDAS, SeaWiFS Project, and NRL).

Product	Description
rrs_*nnn*	remote sensing reflectance at *nnn* nm
nLw_*nnn*	normalized water−leaving radiance at *nnn* nm

Lw_*nnn*	water−leaving radiance at *nnn* nm
Lr_*nnn*	Rayleigh radiance at *nnn* nm
La_*nnn*	aerosol radiance at *nnn* nm
TLg_*nnn*	TOA glint radiance at *nnn* nm
tLf_*nnn*	foam (white−cap) radiance at *nnn* nm
Lt_*nnn*	calibrated TOA radiance at *nnn* nm
t_sol_*nnn*	Rayleigh-aerosol transmittance,sun to ground at *nnn* nm
t_sen_*nnn*	Rayleigh-aerosol transmittance,ground to sensor at *nnn* nm
t_oz_sol_*nnn*	ozone transmittance,sun to ground at *nnn* nm
t_oz_sen_*nnn*	ozone transmittance,ground to sensor at *nnn* nm
taua_*nnn*	aerosol optical depth at *nnn* nm
tau_*nnn*	same as taua_*nnn*
angstrom_*nnn*	aerosol angstrom coefficents (*nnn*,865) nm
eps_*nnn*_*lll*	ratio of *nnn* to *lll* single−scattering aerosol radiances
Es_*nnn*	extra-terestrial surface irradiance at *nnn* nm
rhos_*nnn*	surface reflectance at *nnn* nm
t_o2_*nnn*	total oxygen transmittance at *nnn* nm
foq_*nnn*	f/Q correction to nadir at *nnn* nm
a_*nnn*_arnone	total absorption at *nnn* nm using Arnone algorithm
bb_*nnn*_arnone	backscatter at *nnn* nm using Arnone algorithm
b_*nnn*_arnone	total scattering at *nnn* nm using Arnone algorithm
c_*nnn*_arnone	beam attenuation at *nnn* nm using Arnone algorithm
a_*nnn*_carder	total absorption at *nnn* nm using Carder algorithm
aph_*nnn*_carder	phytoplankton absorption at *nnn* nm using Carder algorithm
adg_*nnn*_carder	detris/gelbstuff absorption at *nnn* nm using Carder algorithm
bb_*nnn*_carder	backscatter at *nnn* nm using Carder algorithm
b_*nnn*_carder	total scattering at *nnn* nm using Carder algorithm
c_*nnn*_carder	beam attenuation at *nnn* nm using Carder algorithm
agmod_carder	Modeled value for ag(400) from Carder's algorithm
agdef_carder	Default value for ag(400) from Carder's algorithm
aphmod_carder	Modeled value for aph(675) from Carder's algorithm
aphdef_carder	Default value for aph(675) from Carder's algorithm
c_length_*nnn*_carder	beam attenuation length at *nnn* nm using Carder algorithm
a_*nnn*_qaa	total absorption at *nnn* nm using QAA algorithm
aph_*nnn*_qaa	phytoplankton absorption at *nnn* nm using QAA algorithm
adg_*nnn*_qaa	detris/gelbstuff absorption at *nnn* nm using QAA algorithm
bb_*nnn*_qaa	backscatter at *nnn* nm using QAA algorithm
b_*nnn*_qaa	total scattering at *nnn* nm using QAA algorithm
c_*nnn*_qaa	beam attenuation at *nnn* nm using QAA algorithm
aph_443_stumpf	phytoplankton absorption at 443 nm using Stumpf's algorithm
adg_412_stumpf	DOM and gelbstuff absorption at 412 nm using Stumpf's algorithm
adg_555_stumpf	DOM and gelbstuff absorption at 555 nm using Stumpf's algorithm
a_412_stumpf	total absorption at 412 nm using Stumpf's algorithm
a_555_stumpf	total absorption at 555 nm using Stumpf's algorithm
chl_oc2	chlorophyll−a concentration using OC2 algorithm
chl_oc4	chlorophyll−a concentration using OC4 algorithm
chlor_a	same as chl_oc4
chl_stumpf	chlorophyll−a concentration using Stumpf's algorithm
chl_carder	chlorophyll−a concentration using Carder's algorithm
chl_octsc	chlorophyll−a concentration using the OCTS−C algorithm
chl_nn	chlorophyll−a concentration derived from pig_nn data
chl_ndpi	chlorophyll−a concentration derived from pig_ndpi data
chl_trees	chlorophyll-a using Tree's algorithm

pig_oc2	pigment concentration derived from chl_oc2
CZCS_pigment	same as pig_oc2
pigment_czcs	same as pig_oc2
pigment_seabam	same as pig_oc2
pig_oc4	pigment concentration derived from chl_oc4
pigment	same as pig_oc4
pig_octsc	pigment concentration derived from chl_octsc
pig_nn	pigment concentration using neural network algorithm
pig_ndpi	pigment concentration using normalized diffence pigment index
K_490	diffuse attenuation at 490 nm using 443/555 ratio
k490	same as K_490
K_length_532	diffuse attenuation at 532 nm using 443/555 ratio
K_532	diffuse attenuation at 532 nm using 490/555 ratio
K_532_Mueller	same as K_532
k532	same as K_532
ndvi	normalized difference vegetation index
evi	enhanced vegetation index
depth	water depth index
par	photosynthetically active radiation
aerindex	aerosol index
aer_model_min	minimum bounding aerosol model #
aer_model_max	maximum bounding aerosol model #
aer_model_ratio	model mixing ratio
aer_num_iter	number of aerosol iterations, NIR correction
glint_coeff	glint radiance normalized by solar irradiance
l2_flags	level−2 processing flags
epsilon	retreived epsilon used for model selection
eps_78	same as epsilon
cloud_albedo	cloud albedo at 865 nm
lats	latitudes (-90.0 to 90.0)
longs	longitudes (-180.0 to 180.0)
solz	solar zenith angle
sola	solar azimuth angle
senz	satellite zenith angle
sena	satellite azimuth angle
windspeed	magnitude of wind at 10 meters
zwind	zonal wind speed at 10 meters
mwind	meridional wind speed at 10 meters
windangle	wind direction at 10 meters
water_vapor	precipital water concentration
humidity	relative humidity
pressure	barometric pressure
ozone	ozone concentration
fsol	solar distance correction (1-D, not an image)
smoke	smoke index
N_small_particles	number of small particles using Haltrin's algorithm
N_large_particles	number of small particles using Haltrin's algorithm
N_particles	number of small particles using Haltrin's algorithm
salinity	salinity using Arnone's algorithm
visibility_*nnn*	diver visibility using 6.1/c (McBride)
horiz_vis	horizontal diver visibility using 4.8/c
vert_vis	vertical diver visibility using 4.0/(c+Kd)

OPTIONS

Each of these options listed below must be placed on the command line or in the parameter files as keyword=value pairs. If the enviornment variables (see "ENVIRONMENTAL VARIABLES" below) are defined, then the only required keyword=value pairs are ifile and ofile1. The others have reasonable defaults.

par
: Input parameter file to be used in the specific command mode `msl12, par="pfile"`. The parameter file is a text file containing the user-defined keyword=value pairs, each on a single line.

Input/Output File

MSl12 has the ability to output up to four seperate files each with their own list of products (ofile1, ofile2, ofile3, and ofile4 contain the names of each file, and l2prod1, l2prod2, l2prod3, and l2prod4 contain the list of desired products for each output file). This allows the user to seperate products into various output files.

ifile
: Directory path and filename of the input level−1A data product. An OCTS input must be either a NASDA L1B format file or a SIMBIOS-format L1B file (generated by l1bocts). A MOS input must be MOS-B L1B data. A SeaWiFs input may be either a L1A or L1B file.

ofile1
: Directory path and filename of the output Level−2 HDF file containing the products specified in the defaults file def_l2prod_file or by the **l2prod1** keyword.

def_l2prod_file
: The file containing the default output products by files. (Default=$MSL12_DATA/*sensor*/*sensor*_def_l2prod.dat.)

l2prod1
: The product names (see main description section above) to be output to ofile1.

ofile[2-4]
: Directory path and filename of the output L2 files containing the products specified in the default file def_l2prod_file or by the l2prod[2-4] keyword.

l2prod[2-4]
: The product names (see main description section above) to be output to ofile[2-4].

ofmt *n*
: This keyword defines the type of file output. If set to 0 or 1, a flat binary type of format is written. The default value of 2, creates the standard NASA HDF fileformat. A value of 3 will produce an NRL HDF fileformat.

Input File Parameters

MSl12 has the ability to subsection/subsample the input file and to control the number of control points in each output file. If not specified, MSl12 will work on the entire input file.

spixl
: Starting pixel to be processed (default=1)

epixl
: End pixel to be processed (default=0, meaning the last pixel in the scan line)

dpixl
: Pixel subsampling interval (default=1)

sline Starting line number to be processed (default=1)

eline Ending line number to be processed (default=0, meaning the last line in file)

dline Line subsampling interval (default=1)

ctl_pt_incr
Control-point pixel increment for lon/lat arrays (0=optimize, default=8)

Calibration Control Options

These keywords control the calibration of the input data.

calfile Directory path and filename of the input calibration table file. The default is $CAL_HDF_PATH.

vcal_opt
Control for calibration modification: 0 - use gain, offset from the calibration table 1 - use the gain (as multipliers of the existing gains in the table file) from the input (gain) and offset from the calibration table 2 - use offsets from the input (offset) and the gains from the calibration table 3 - use both gain (as multipliers of the existing gains in the table file) and offset from the input (gain & offset)

gain Calibration gain factors to multiply the gain values read from the $MSL12_DATA/*sensor*/*sensor*_table.dat file. Defaults to [0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0].

offset Calibration gain offset factors to substitute for values read from the calibration table. Defaults to [1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0].

Ancillary Input Files

These keywords control the ancilliary input files. If the environment variables have been set (see ‘‘ENVIRONMENTAL VARIABLES’’), then these keywords have reasonable defaults. If they are not set, then the user *must* define these: land, water, met1, and ozone1.

land Directory path and filename of the land-mask input file. Default is $MSL12_DATA/landmask.dat

water Directory path and filename of the shallow water mask input file used for setting the l2_flags bit to indicate shallow water (defined as 30m) areas (McClain et al., 1995). Default is $MSL12_DATA/watermask.dat

met1 Directory path and filename of the climatological product or the near−real−time (NRT) meteorological ancillary data product available for the nearest time preceding the time of ifile product’s first scan line. If met1 is the climatological file, then met2 and met3 will not be used, otherwise, see met2 for logic. Default is $MSL1_DATA/CLIMATOLOGY.MET.

met2 Directory path and filename of the NRT meteorological ancillary data product available for the nearest time following the time of ifile product’s first scan line’s. If met2 is not specified (null) and met1 is a NRT product, then met2 will be set to met1. If met1 <> met2 and the scan line’s date and time, fall between the times of met1 and met2, get_ancillary will use met1 and met2 to generate the interpolated meteorological values (if the scan line’s date and time fall before those of met1, an error occurs). If met1 = met2 and the scan line’s date and time fall before met2, get_ancillary will use only met2 to generate the meteorological values. If met2 <> met3 and the scan line's date and time fall between the times of met2 and met3, get_ancillary will use met2 and met3 to generate the interpolated meteorological values (if the scan line's date and time fall after those of met3, an error occurs). If met2 = met3 and the scan line's date and time fall after met2, get_ancillary will use only met2 to generate the meteorological values.

met3 Directory path and filename of the NRT meteorological ancillary data product for the nearest time following the time of ifile product's last scan line. If met3 is not specified (null) and met1 is a NRT product, then met3 will be set to met2 and the logic specified in met2 will be applied.

ozone1 Directory path and filename of the climatological product or the NRT ozone ancillary data product available for the nearest time preceding the time of ifile product's first scan line. If ozone1 is the climatological file, then ozone2 and ozone3 will not be used, otherwise see ozone2 for logic. (For TOVS data, the center point time is used to represent the time of that product.) Defaults to $MSL12_DATA/CLIMATOLOTY.OZONE.

ozone2 Directory path and filename of the NRT ozone ancillary data product available for the nearest time following the time of ifile product's first scan line's. If ozone2 is not specified (null) and ozone1 is a NRT product, then ozone2 will be set to ozone1. If ozone1 <> ozone2 and the scan line's date and time, fall between the times of ozone1 and ozone2, get_ancillary will use ozone1 and ozone2 to generate the interpolated ozone values (if the scan line's date and time fall before those of ozone1, an error occurs). If ozone1 = ozone2 and the scan line's date and time fall before ozone2, get_ancillary will use only ozone2 to generate the ozone values. If ozone2 <> ozone3 and the scan line's date and time fall between the times of ozone2 and ozone3, get_ancillary will use ozone2 and ozone3 to generate the interpolated ozone values (if the scan line's date and time fall after those of ozone3, an error occurs). If ozone2 = ozone3 and the scan line's date and time fall after ozone2, get_ancillary will use only ozone2 to generate the ozone values. (For TOVS data, the center point time is used to represent the time of that product.)

ozone3 Directory path and filename of the NRT ozone ancillary data product for the nearest time following the time of ifile product's last scan line. If ozone3 is not specified (null) and ozone1 is a NRT product, then ozone3 will be set to ozone2 and the logic specified in ozone2 will be applied. (For TOVS data, the center point time is used to represent the time of that product.)

Algorithm control options

These keywords modify and/or select the algorithms used to correct the input data or change certain thresholds used for various tests.

carder_opt

This option selects the parameters to use for Carder's algorithm. 0=global parameters (default), 1=unpackaged parameters, 2=packaged parameters 3=hyper-packaged parameters, 4=use an remote sensing reflectance filter to select either global, unpackaged, or packaged paramters, 5=use the value read from the parameter file.

carder_version

This option selects the version of the Carder algorithm to use. 0=version 1.3, 1=version 1.33, 2=version 1.4 Version 1.33 adds the estimates for default aph_mod and ag_mod and blends these with the modeled values. Version 1.4 add the hyperpacked parameters.

carder_iter
: This option turns on the s iteration in Carder's algorithm. 0=off, 1=on.

carder_gain
: Calibration gain factors to multiply the remote sensing reflectance values passed into the carder function. Defaults to [1.0,1.0,1.0,1.0,1.0,1.0].

filter_opt
: Option for filtering the L1B data with the method specified in filter_file. 1=apply filtering , 0=do not apply filtering. (Default for OCTS=1, Default for others=0).

filter_file
: Directory path and filename of the filter file that contains the filter method and information to be applied to the L1B data when **filter_opt** is set to 1. (Default=$MSL12_DATA/*sensor*/*sensor*_filter.dat).

outband_opt
: SeaWiFs out-of-band corrections. 0 = no correction, 1 = no Lw correction, 2 = full correction. (Default = 1 for SeaWiFs, 0 for all others)

oxaband_opt
: SeaWiFs/OCTS 764nm band Oxygen correction. 0 = off, 1 = on. (Default = 1 for SeaWiFs and OCTS, 0 for others)

glint_opt
: Correct for residual glint radiance. 0 = off, 1 = on. (Default = 0)

aer_opt
: Option for aerosol calculation mode. The default is -3.

Value	Description
1	Multi-scattering with fixed model (Oceanic, 90% humidity)
2	Multi-scattering with fixed model (Oceanic, 99% humidity)
3	Multi-scattering with fixed model (Maritime, 50% humidity)
4	Multi-scattering with fixed model (Maritime, 70% humidity)
5	Multi-scattering with fixed model (Maritime, 90% humidity)
6	Multi-scattering with fixed model (Maritime, 99% humidity)
7	Multi-scattering with fixed model (Coastal, 50% humidity)
8	Multi-scattering with fixed model (Coastal, 90% humidity)
9	Multi-scattering with fixed model (Coastal, 99% humidity)
10	Multi-scattering with fixed model (Tropospheric - 50% humidity)
11	Multi-scattering with fixed model (Tropospheric - 90% humidity)
12	Multi-scattering with fixed model (Tropospheric - 99% humidity)
0	Single-scattering white aerosols (CZCS algorithm)
-1	Multi-scattering with 765/865 model selection (default for OCTS)
-2	Multi-scattering with 670/865 model selection (default for MOS)
-3	Multi-scattering with 765/865 model selection and Siegel NIR iteration
-4	Multi-scattering with 670/865 model selection and Siegel NIR iteration
-100	Multi-scattering with 765/865 model selection and Arnone NIR iteration
-101	Multi-scattering with 765/865 model selection and Arnone NIR (+aph) iteration
-102	Multi-scattering with 765/865 model selection and Arnone NIR (+adg) iteration
-103	Multi-scattering with 765/865 model selection and Arnone NIR (+aph+adg) iteration
-110	Multi-scattering with 765/865 model selection and Arnone NIR and Stumpf 412 iteration

-111	Multi-scattering with 765/865 model selection and Arnone NIR (+aph) and Stumpf 412 iteration
-112	Multi-scattering with 765/865 model selection and Arnone NIR (+adg) and Stumpf 412 iteration
-113	Multi-scattering with 765/865 model selection and Arnone NIR (+aph+adg) and Stumpf 412 iteration
-200	Multi-scattering with 765/865 model selection and Stumpf NIR (+aph+adg)
-201	Multi-scattering with 765/865 model selection and Stumpf NIR (+aph+adg) with 0.5% Lt412 adjustment
-202	Multi-scattering with 765/865 model selection and Stumpf NIR (+aph+adg) with 1.0% Lt412 adjustment
-203	Multi-scattering with 765/865 model selection and Stumpf NIR (+aph+adg) with 1.5% Lt412 adjustment
-250	Multi-scattering with 765/865 model selection and Stumpf NIR (+aph+adg) and Stumpf 412 iteration
-251	Multi-scattering with 765/865 model selection and Stumpf NIR (+aph+adg) and Stumpf 412 iteration with
-252	Multi-scattering with 765/865 model selection and Stumpf NIR (+aph+adg) and Stumpf 412 iteration with
-253	Multi-scattering with 765/865 model selection and Stumpf NIR (+aph+adg) and Stumpf 412 iteration with
-300	Multi-scattering with 765/865 model selection and MUMM NIR

aer_iter_min
The minimum number of aerosol iterations (Default = 1).

aer_iter_max
The maximum number of aerosol iterations (Default = 10 for any NIR algorithm).

tau_a Aerosol optical thickness at 865 nm for fixed model. If tau_a > 0 and aer_opt > 0, then the input tau_a will be used to derive aerosol reflectance.

qaa_opt
Blending option for QAA algorithm. 0 is for blending the QAA-555 and QAA-640 absorptions and backscattering, 1 is for no blending, use QAA-555 results only, and 2 is for no blending, use QAA-670 results only.

qaa_S Define the S parameter to use in the QAA algorithm. Default is 0.015.

sunzen Solar zenith angle in degrees; threshold for setting the l2_flags bit to indicate large solar zenith angles (McClain et al., 1995). Default is 65.0.

satzen Spacecraft zenith angle in degrees; threshold for setting the l2_flags bit to indicate large satellite zenith angles (McClain et al., 1995). Default is 56.0.

albedo Cloud albedo for band 8 in percent; threshold for setting the l2_flags bit to indicate clouds or ice (McClain et al., 1995).

absaer Absorbing aerosol threshold on aerosol index. Default is 0.5.

glint_thresh
Sun glint threshold (fraction of F0(865)); used in calculations for setting the l2_flags bit to indicate sun glint (McClain et al., 1995).

tauamax
Maximum 865 aerosol optical depth used for setting the l2_flags bit #5. (Default=0.3)

epsmin Minimum epsilon to trigger atmospheric correction failure flag (default is 0.65).

epsmax Maximum epsilon to trigger atmospheric correction failure flag (default is 1.35).

nlwmin Minimun nlw(555) used for setting the l2_flags bit #15. (Default=0.15).

wsmax Windspeed limit on white-cap correction. (Default=8.0 m/s).

sl_frac Lt 865 threshold for stray-light correction. (SeaWiFs only.) (Default = 0.25)

sl_pixl Number of LAC pixels over which stray−light correction is applied. 0 = no correction, -1 = program defaults (8 for GAC, 3 for LAC)

Masking keywords

These keywords select flags to be used as masks. A mask is a special flag that will cease execution on the pixel which passes the flag. For example, maskland will skip all land pixels.

maskland
Mask out land pixels: 0=off, 1=on. (Default=1).

maskbath
Mask out shallow water pixels: 0=off , 1=on. (Default=0).

maskcloud
Mask out cloud or ice pixels: 0=off, 1=on. (Default=1).

maskglint
Mask out sun glint pixels: 0=off, 1=on. (Default=1).

masksunzen
Mask out large solar zenith angle pixels: 0=off, 1=on. (Default=0).

masksatzen
Mask out large sensor zenith angle pixels: 0=off, 1=on. (Default=0).

maskhilt
Mask out pixels for which total radiance was greater than knee value: 0=off, 1=on. (Default=1).

maskstlight
Mask out stray light contaminated pixels: 0=off, 1=on. (Default=1).

Debugging controls

station_input
The name of an input station file. The format is UNIX text file with three columns. The first two are the sample and line location of the desired pixel and the last is an ASCII station name field. During each iteration of the atmosopheric correction information will be dumped to an output file designated by station_output

station_output

The name of an output station file. This will receive the data dumps for each station for each iteration of the atmospheric correction algorithm.

ENVIRONMENTAL VARIABLES

MSL12_DATA

Root directory for atmospheric correction data files. If this environmental variable is not set, then several ancilliary input file keywords will have to be defined. See “ANCILLIARY INPUT FILES” above.

CAL_HDF_PATH

The directory and pathname of the sensor calibration file. May be undefined if **calfile** keyword has been set.

CARDER_FILE

The directory and path of the parameter file for Carder’s algorithm. May be undefined if no products using Carder’s algorithm have been selected.

EXAMPLES

This the minimum command line execution (assumes the enviornmental variables have been set as shown):

```
$ export MSL12_DATA=/home/aps/aps_v2.6/data
$export CAL_HDF_PATH=$MSL12_DATA/seawifs/cal/SEAWIFS_SENSOR_CAL.TBL
$ MSl12 ifile=S2000148173615.L1A_HNAV ofile1=S2000148173615.L2_HNAV
```

In this example, the user has selected to output the remote sensing reflectance data into one file, the bio-optical products K_490 and chl_oc4 in another, and some of the Arnone IOP products in a third. It also turns off the masking of high Lt values and increases the cloud and ice threshold to 1.5.

```
$ MSl12 ifile=S2000148173615.L1A_HNAV ofile1=S2000148173615.L2.rrs \
     l2prod1="rrs_412 rrs_443 rrs_490 rrs_510 rrs_555 rrs_670" \
     ofile2=S2000148173615.L2.bio l2prod2="chl_oc4 K_490" \
     ofile3=S2000148173615.L2.iop l2prod3="a_443_arnone bb_555_arnone" \
     albedo=1.5 maskhilt=0
```

NAME

swfL1bgen − Generate Level−1B file from Level−1A

SYNOPSIS

swfL1bgen ifile=ifile ofile=ofile [straylight=straylight1] [outband=outband] [calmod_flg=calmod_flg] [calmod_gain=calmod_gain] [calmod_off=calmod_off] [calhdf=calhdf]

− or −

swfL1bgen par=par

DESCRIPTION

This program reads in SeaWiFS Level−1A file (HRPT, LAC, or GAC) and generates a Level−1B file by applying the sensor calibration at the same resolution and location as the Level−1A data. The detail information about Level−1B HDF file format is available in "SeaWiFS Non-Archive Product Specification".

OPTIONS

ifile Directory path and filename of the input Level−1A data product.

ofile Directory path and filename of the output Level−1B data product.

calhdf Directory path and filename of the calibration table file.

straylight

First value: if 0, stray-light correction is not applied to l1b_data of get_l1a_record; else, the value equals the (rounded integer) number of along-scan LAC pixels on either side of a bright target to correct l1b_data for stray light. (For GAC data, that number is divided by four and rounded up to the nearest integer to represent the number of along-scan GAC pixels to correct.) Currently, the SeaWiFS project uses 3 for LAC and HRPT data, and 8 for GAC data. In SeaDAS, a 999 value is used for the default. In this case, the program will check the input file and use 3 for LAC and HRPT data, and 8 for GAC data.

Second value: fraction of L typical for band 8 to use as a threshold in adjacent pixels' radiance difference for detecting the edge of a bright target in the stray-light correction algorithm; ignored if first value is negative.

outband

if 0 or negative, out-of-band correction is not applied to l1b_data of get_l1a_record; else, l1b_data is the index of the out-of-band correction method to use.

calmod_flg

Control for calibration modification:
0 - use gain, offset from the calibration table
1 - use the gains from the input (calmod_gain) and offset from the calibration table
2 - use offsets from the input (calmod_off) and the gains from the calibration table
3 - use both gain and offset from the input (calmod_gain & calmod_off)

calmod_gain

Calibration gain factors to substitute for values read from the calibration table.

calmod_off
Calibration gain offset factors to substitute for values read from the calibration table.

par
Input parameter file to be used in the specific command mode 'l1bgen, par="pfile"'. The parameter file may be generated in the interactive mode and used in the command mode.

SEE ALSO

swfL1agen(1) **swfL2gen**(1) **swfCase2**(1)

NAME

mosInfo − query information about a MOS Level−1A file

SYNOPSIS

mosInfo [option] mosFile

DESCRIPTION

This program is used to dump information about a MOS data file. With no options the program will print out a series of parameters. A single parameter can be single with the option. The options are succint as they were designed with shell scripting in mind.

OPTIONS

-day Day of month of input file.

-doy Day of year of input file.

-hour 2-digit time (HH) of input file.

-min 2-digit time (MM) of input file.

-month 3-character month of input file. Months are "jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"

-name Generate a file name in the following format as mos.YYYY.MMDD.HHMM. This is a short cut version of using −sat, −year, −doy, and −time.

-sat 3-character satellite name. Names is "mos".

-time 4-digit time (HHMM) of input file.

-type Character code for datatype: "LAC", "GAC", "HRPT"

-year 2-digit year of input file.

--version
Print out version and exit

EXAMPLES

```
$ mosInfo M1998291133955.L1A_GAC
Filename:        M1998291133955.L1A_GAC
Starting Time:   10/18/1998 13:39, 291
Ending Time:     10/18/1998 14:19, 291
Satellite:       mos
Datatype:        GAC
Total Scans:
$

$ mosInfo −year M1998291133955.L1A_GAC
1998
```

```
$
```

Here is how a Bourne shell script function might use **mosInfo** to set the name of the output files from:

```
 set_name()
{
    sat=`mosInfo -sat $1`
    yr=`mosInfo -year $1`
    jday=`mosInfo -doy $1`
    time=`mosInfo -time $1`
    file=M$yr$jday$time.L1A_HNAV
}
```

NAME

mosScripts – standard MOS processinig scripts

DESCRIPTION

mosScripts provides Bourne shell scripting functions to process MOS data from a NASA MOS Level−1B file. The mosProcess script function is the main user interface and provides for the standard processing steps. These steps are:

1 Verify input file is NASA MOS Level−1B file using filefmt program.
3 Get time information from file using mosInfo program.
4 Run MSl12 program to atmospherically correct raw data
7 For a large product array, tile and compress the data.
8 If user had defined $GifList, generate browse images for selected products.
9 If user has defined $MosPostProcess, it is now called. (Normally, not defined.)
10 Store HDF product file in the $DATA_BASE and the browse image in the $IMAG_BASE.

SETUP

A minimalist executable script for processing MOS data must source both the apsScripts and mosScripts located in the APS bin directory, call the script function mosProcess passing it the name of the file name containing the MOS data and define at least two script variables. Many other variables can be optionally set to modify the *normal* mode of operations. These must be set prior to the call to mosProcess.

```
#!/bin/sh
test -z "$AUTO_DIR" && return 1
MapName=ChesapeakeBay_MOS
MapExt=GOM
mosProcess $1
```

The script must be placed in the `$AREAS_PROC` directory which is normally the directory `areas` located in the main APS directory. The script must have execution permissions.

REQUIRED VARIABLES

These variables are required to process an area. They provide the script an area to process. Most of the remaining variables have defaults which can be overridden. They are described in the next section.

MapName

This is the name of the image map stored in the file $MapFile.

MapExt

This is a string that is appended to the Level−3 file which is written to the database. Usually it is a three character extension all uppercase.

OPTIONAL VARIABLES

The following sections are variables that have defaults which the user can override to change the behaviour of the default processing. They are grouped together by subject.

Product Selection

ProdList

This is a space delimetered list of products to be written to the output data base. The following products are available:

a_412_Arnone	Total Absorbtion at 412nm using Arnone
a_443_Arnone	Total Absorbtion at 443nm using Arnone
a_490_Arnone	Total Absorbtion at 490nm using Arnone
a_510_Arnone	Total Absorbtion at 510nm using Arnone
a_555_Arnone	Total Absorbtion at 555nm using Arnone
a_670_Arnone	Total Absorbtion at 670nm using Arnone
bb_412_Arnone	Backscattering at 412nm using Arnone
bb_443_Arnone	Backscattering at 443nm using Arnone
bb_490_Arnone	Backscattering at 490nm using Arnone
bb_510_Arnone	Backscattering at 510nm using Arnone
bb_555_Arnone	Backscattering at 555nm using Arnone
bb_670_Arnone	Backscattering at 670nm using Arnone
b_412_Arnone	Total Scattering at 412nm using Arnone
b_443_Arnone	Total Scattering at 443nm using Arnone
b_490_Arnone	Total Scattering at 490nm using Arnone
b_510_Arnone	Total Scattering at 510nm using Arnone
b_555_Arnone	Total Scattering at 555nm using Arnone
b_670_Arnone	Total Scattering at 670nm using Arnone
c_412_Arnone	Beam Attenuation at 412nm using Arnone
c_443_Arnone	Beam Attenuation at 443nm using Arnone
c_490_Arnone	Beam Attenuation at 490nm using Arnone
c_510_Arnone	Beam Attenuation at 510nm using Arnone
c_555_Arnone	Beam Attenuation at 555nm using Arnone
c_670_Arnone	Beam Attenuation at 670nm using Arnone
a_Arnone	Absorption for all six bands.
bb_Arnone	Backscattering for all six bands.
Arnone	Absorption and Backscattering for all six bands.
a_412_Carder	Total Absorbtion at 412nm using Carder
a_443_Carder	Total Absorbtion at 443nm using Carder
a_490_Carder	Total Absorbtion at 490nm using Carder
a_510_Carder	Total Absorbtion at 510nm using Carder
a_555_Carder	Total Absorbtion at 555nm using Carder
a_670_Carder	Total Absorbtion at 670nm using Carder
bb_412_Carder	Backscattering at 412nm using Carder
bb_443_Carder	Backscattering at 443nm using Carder
bb_490_Carder	Backscattering at 490nm using Carder
bb_510_Carder	Backscattering at 510nm using Carder
bb_555_Carder	Backscattering at 555nm using Carder
bb_670_Carder	Backscattering at 670nm using Carder
b_412_Carder	Total Scattering at 412nm using Carder
b_443_Carder	Total Scattering at 443nm using Carder
b_490_Carder	Total Scattering at 490nm using Carder
b_510_Carder	Total Scattering at 510nm using Carder
b_555_Carder	Total Scattering at 555nm using Carder
b_670_Carder	Total Scattering at 670nm using Carder

c_412_Carder	Beam Attenuation at 412nm using Carder
c_443_Carder	Beam Attenuation at 443nm using Carder
c_490_Carder	Beam Attenuation at 490nm using Carder
c_510_Carder	Beam Attenuation at 510nm using Carder
c_555_Carder	Beam Attenuation at 555nm using Carder
c_670_Carder	Beam Attenuation at 670nm using Carder
aph_412_Carder	Absorbtion due to phytoplankton at 412nm
aph_443_Carder	Absorbtion due to phytoplankton at 443nm
aph_490_Carder	Absorbtion due to phytoplankton at 490nm
aph_510_Carder	Absorbtion due to phytoplankton at 510nm
aph_555_Carder	Absorbtion due to phytoplankton at 555nm
aph_670_Carder	Absorbtion due to phytoplankton at 670nm
adg_412_Carder	Absorbtion due to detritis and gelbstuff at 412nm
adg_443_Carder	Absorbtion due to detritis and gelbstuff at 443nm
adg_490_Carder	Absorbtion due to detritis and gelbstuff at 490nm
adg_510_Carder	Absorbtion due to detritis and gelbstuff at 510nm
adg_555_Carder	Absorbtion due to detritis and gelbstuff at 555nm
adg_670_Carder	Absorbtion due to detritis and gelbstuff at 670nm
chlor_a_Carder	Chlorophyll-a concentration
a_Carder	Absorption for all six bands.
adg_Carder	Absorption due to detritis and gelbstuff for all six bands.
aphi_Carder	Absorption due to phytoplankton for all six bands.
bb_Carder	Backscattering for all six bands.
Carder	Same as ProdList="a_Carder adg_Carder aphi_Carder bb_Carder chlor_a_Carder"
rrs_412	Remote Sensing Reflectance at 412
rrs_443	Remote Sensing Reflectance at 443
rrs_490	Remote Sensing Reflectance at 490
rrs_510	Remote Sensing Reflectance at 510
rrs_555	Remote Sensing Reflectance at 555
rrs_670	Remote Sensing Reflectance at 670
rrs_765	Remote Sensing Reflectance at 765
rrs_865	Remote Sensing Reflectance at 865
K_490_SeaDAS	Diffuse Attenuation at 490nm using 443/555
K_532_SeaDAS	Diffuse Attenuation at 532nm using 490/555
chlor_a_SeaDAS	Chlorophyll−a concentration using OC2
chlor_a_Stumpf	Chlorophyll−a concentration using Stumpf
pigments_SeaDAS	Pigments
l2_flags	Flags
tau_865	Tau at 865 nm
epsilon	Epsilon
albedo_865	Percent albedo at 865nm
glint_865	Glint at 865 nm
foam_865	Foam at 865 nm
latitude	Latitude
longitude	Longitude

solar_zenith	Solar Zenith Angle
solar_azimuth	Solar Azimuth Angle
sat_zenith	Satellite Zenith Angle
sat_azimuth	Satellite Azimuth Angle
adg_412_Stumpf	Absorption due to Gelbstoff
aphi_443_Stumpf	Absorption due to Phytoplankton
a_555_Stumpf	Total Absorption at 555
a_412_Stumpf	Total Absorption at 412
adg_555_Stumpf	Absorption due to Gelbstoff
Rt_412	Total Reflectance at 412
Rt_443	Total Reflectance at 443
Rt_490	Total Reflectance at 490
Rt_510	Total Reflectance at 510
Rt_555	Total Reflectance at 555
Rt_670	Total Reflectance at 670
Rt_765	Total Reflectance at 765
Rt_865	Total Reflectance at 865
Rr_412	Rayleigh Reflectance at 412
Rr_443	Rayleigh Reflectance at 443
Rr_490	Rayleigh Reflectance at 490
Rr_510	Rayleigh Reflectance at 510
Rr_555	Rayleigh Reflectance at 555
Rr_670	Rayleigh Reflectance at 670
Rr_765	Rayleigh Reflectance at 765
Rr_865	Rayleigh Reflectance at 865
Ra_412	Aerosol Reflectance at 412
Ra_443	Aerosol Reflectance at 443
Ra_490	Aerosol Reflectance at 490
Ra_510	Aerosol Reflectance at 510
Ra_555	Aerosol Reflectance at 555
Ra_670	Aerosol Reflectance at 670
Ra_765	Aerosol Reflectance at 765
Ra_865	Aerosol Reflectance at 865
zone_w	Zonal Winds
merid_w	Meridian Winds
humidity	Humidity
pressure	Pressure
ozone	Ozone

For example, at NRL we process the 33 products for each scene. Here is how the product list is defined in our script `SwfGulfofMexico`.

```
# Name all desired products

ProdList="rrs_412 rrs_443 rrs_490 rrs_510 rrs_555 rrs_670"
ProdList="$ProdList K_532_Mueller albedo_865 l2_flags"
ProdList="$ProdList chlor_a_SeaDAS chlor_a_Stumpf"
ProdList="$ProdList chlor_a_Carder"
ProdList="$ProdList a_Arnone"
```

```
ProdList="$ProdList a_Carder"
ProdList="$ProdList adg_412_Carder aph_443_Carder"
ProdList="$ProdList adg_412_Stumpf aph_443_Stumpf"
ProdList="$ProdList bb_443_Arnone bb_555_Arnone"
ProdList="$ProdList bb_443_Carder bb_555_Carder"
ProdList="$ProdList c_670_Carder"
```

Region
This variable will be used to create the default data base directories. By default it set to `$MapName`.

MosDataBase
This variable is used to indicate the location of the image data base for the generated product file. By default, it is set to:

`$DataBase/$Level/$Sensor/$Version/$Region/$Year/$Month`.

where,
`$DataBase` is defined in the `aps.conf` file and represents the top directory of the data base.
`$Level` is set to the string "lvl3" by mosInit.
`$Sensor` is set to the string "mos" by mosInit.
`$Version` is set to "2.4" by mosInit.
`$Region` is set to "$MapName" by mosInit.
`$Year` and `$Month` are set by `mosProcess` based on the input file.

The user can override `$SwfDataBase` since it is evaluated by the shell prior to use. For example, if the line:

`SwfDataBase="\$DataBase/seawifs/\$Year"`

is set in the areas script and we assume that `$DataBase` is set to `/data` and that for a particular file `$Year` has been set to 1999, then the product file will be moved to `/data/seawifs/1999`. Note that to use the variables, the user must "escape" the '$' by inserting a "\".

CmpOpt
This can be defined by the user to select the type of compression program to call for the output product file before it is moved to $SwfDataBase. This option can be set to: "gzip", "compress", "bzip2" or "none". Only set CmpOpt to a compression type that is available on user's machine. *Note:* If the user has defined XNumChunks or that variable is defined by mosInit, then the HDF file will automatically be internally compressed and this variable will have no effect.

mosDebug
If defined this variable will cause the script functions to call 'set -x' within each script function. This will have the effect of printing out each step as it is executed.

MapFile
Name of file containing image map file. Defaults to $AutoData/maps.hdf

MinPixels, MinLines
Used to set the minimum pixels/lines that must be extracted from the MOS file by swfExtract to continue processing. These are used to insure that enough of the input file covers the area of interest. By default these are not defined and, therefore, no check is performed.

XNumChunks, YNumChunks

These are used to define the number of "chunks" in each direction which will be created by imgReformat program for the product files. By default, a large level−3 file will be rewritten in chunks (or "tiles") with each chunk being compressed. A chunk will be no larger than 640 by 640. So, if the map image is 2430 by 1810, then imgReformat will create a total of 12 chunks (4 across and 3 down). A 1500 by 1500 map image will be divided into 9 chunks (3 across and 3 down). A 600 by 300 map image will NOT be chunked.

FileExt

Optional extension to L2 files which can be used to prevent filename clashes. Normally, not defined.

MOS Processing Parameters

CalFile

Name of calibration file to be used during Level−2 processing. Defaults to $AUTO_DATA/seawifs/SEAWIFS_SENSOR_CAL.TBL-199909-time_dep_7jun99.

LandMask

Name of landmask file to be used during Level−2 processing. Defaults to $AutoData/seawifs/landmask.dat

WaterMask

Name of watermask file to be used during Level−2 processing. Defaults to $AutoData/seawifs/watermask.dat

MosAtmOpts

This controls the atmospheric algorithm used. By default, it is set to "stumpf". It can also be removed to get the default atmospheric correction. If set to "arnone", then the Arnone NIR iteration alone is run. If "aphi" is added, for example, `SwfAtmOpts="arnone aphi"` then the Arnone NIR iteration with the aphi adjustment is run. The "aphi" option will have not effect if added to stumpf or run on its own. You can use both "stumpf" and "arnone" at the same time.

MosParamFile

Name of file containing additional options to the MSl12 program. These should *not* contain the following options: par=, in=, out=, outqc=, calhdf= (use CalFile instead), landmask= (use LandMask instead), watermask= (use WaterMask instead), met1=, met2=, met3= (use SwfMetDir, SwfMetFile instead), ozone1=, ozone2=, ozone3= (use SwfOzoneDir, SwfOzoneFile instead). All others should present no problems. The user's options located in the file pointed to by SwfParamFile are appended the to parameter file automatically generated by swfScripts.

MosOzoneDir, MosOzoneFile

Normally, these are undefined and will cause the climatology ozone file located in the data/seawifs directory to be used.

To use the NRT ozone data, the user must download it from the Goddard DAAC. The variable MosOzoneDir points to the location of the NRT ozone file(s). If MosOzoneFile is not set, then swfNRL will use the input file's date to build the appropriate file name. If this file had been compressed using gzip or compress, it is uncompressed. If MosOzoneFile is set, it must be an uncompressed file -- swfNRL sends it straight to l2gen, which will die if the file is incorrect.

If MosOzoneDir is undefined, or the file $MosOzoneDir/$MosOzoneFile is not a regular file, then climatology file will be used.

MosMetDir, MosMetFile
Like MosOzoneDir and MosOzoneFile defined above but for the MET data.

Browse Image Variables

GifList This is a list of whitespace delimetered products which are converted to browse images. The products in this list must also be present in the $ProdList variable. By default, no browse images are created. That is, GifList is not defined.

GifDir This variable is used to indicate the location of for the browse images. By default, it is set to:
`$ImagBase/$Level/$Sensor/$Version/$Region/$Year/$Month`.
where,
`$ImagBase` is set in the `aps.conf` file and represents the top directory of the browse data base.
`$Level` is set to the string "lvl3" by `mosInit`.
`$Sensor` is set to the string "mos" by `mosInit`.
`$Version` is set to "2.4" by `mosInit`.
`$Region` is set by the user in the area script.
`$Year` and `$Month` are set by `mosProcess` based on the input file.

The user can override `$GifDataBase` since it is evaluated by the shell prior to use. For example, if the line:

```
GifDataBase="\$ImagBase/browse/\$Year"
```

is set in the areas script and we assume that $ImagBase is set to `/data` and that for a particular file `$Year` has been set to 1999, then the browse image will be moved to `/data/browse/1999`.

${prod}_GifScaling
It defines the user's desired output scaling for the browse images. For example, to produce an chlorophyll−a browse image that is 300 pixels by 400 lines and has a data range from 0.01 to 10.0 using a log10 scale, add the line:

```
chlor_a_SeaDAS_GifScaling="-f log10 -r 0.01,10.0 -R 20,199 -s
300,400"
```

in the areas script. See **imgBrowse**(1) for more information. *Note:* The output range is set to 20 and 199 because the swfMakeGif file script uses NSIPS files for overlays and colortables by default.

By default, the following scaling is defined by swfScripts.

channel	function	min	max
rrs_412	linear	-0.005	0.015
rrs_443	linear	-0.005	0.015
rrs_490	linear	-0.005	0.015
rrs_510	linear	-0.005	0.015
rrs_555	linear	-0.005	0.015
rrs_670	linear	-0.005	0.015
rrs_765	linear	-0.005	0.015
rrs_865	linear	-0.005	0.015
K_490_SeaDAS	log10	0.01	2.0
K_532_SeaDAS	log10	0.02	2.0

K_532_Mueller	log10	0.01	2.0
pigments_SeaDAS	log10	0.01	64.0
chlor_a_Carder	log10	0.01	45.0
chlor_a_SeaDAS	log10	0.01	45.0
chlor_a_Stumpf	log10	0.01	15.0
aph_443_Stumpf	log10	0.002	1.0
adg_412_Stumpf	log10	0.001	10.0
adg_555_Stumpf	log10	0.0001	0.0175
a_412_Stumpf	log10	0.01	1.5
a_555_Stumpf	log10	0.01	1.5
a_412_Arnone	log10	0.005	1.5
a_443_Arnone	log10	0.001	2.0
a_490_Arnone	log10	0.005	1.5
a_510_Arnone	log10	0.005	1.5
a_555_Arnone	log10	0.005	1.5
a_670_Arnone	linear	0.4	0.9
a_765_Arnone	linear	0.4	4.3
bb_412_Arnone	log10	0.0005	0.5
bb_443_Arnone	log10	0.0005	0.5
bb_490_Arnone	log10	0.0005	0.5
bb_510_Arnone	log10	0.0005	0.5
bb_555_Arnone	log10	0.0005	0.2
bb_670_Arnone	log10	0.0005	0.5
bb_765_Arnone	log10	0.0005,0.5	
a_412_Carder	log10	0.01	1.5
a_443_Carder	log10	0.01	1.5
a_490_Carder	log10	0.01	1.5
a_510_Carder	log10	0.01	1.5
a_555_Carder	log10	0.01	1.5
a_670_Carder	log10	0.01	1.5
aph_412_Carder	log10	0.00005	0.5
aph_443_Carder	log10	0.001	1.0
aph_490_Carder	log10	0.00005	0.5
aph_510_Carder	log10	0.00005	0.5
aph_555_Carder	log10	0.00005	0.5
aph_670_Carder	log10	0.00005	0.5
adg_412_Carder	log10	0.010	1.0
adg_443_Carder	log10	0.005	0.5
adg_490_Carder	log10	0.001	0.18
adg_510_Carder	log10	0.0007	0.12
adg_555_Carder	log10	0.0001	0.0175
adg_670_Carder	log10	0.00001	0.00175
bb_412_Carder	log10	0.0005	0.5
bb_443_Carder	log10	0.0005	0.5
bb_490_Carder	log10	0.0005	0.5

bb_510_Carder	log10	0.0005	0.5
bb_555_Carder	log10	0.0005	0.2
bb_670_Carder	log10	0.0005	0.5

Note: For the other products (those list in ProdList above but not here), the user must provide the corresponding variables as there are no defaults.

${prod}_Overlay

To apply an overlay, the user may set this variable. By default, it is defined to be $AutoData/areas/$MapName/w_${Sensor}_$prod.pic. This file is overlayed over the image file. See **nsOverlay**(1) for more information.

GifLabelOpts, ${prod}_GifLabelOpts, and ${prod}_GifLabel

To add a label, the user must define at least GifLabelOpts. This will consist of three space delimetered values containing the x−location, y−location, and color index of the label. For an individual product, the user may define ${prod}_GifLabelOpts. If not defined, it will default to GifLabelOpts. The label to be written at that location is defined by ${prod}_GifLabel. If not defined, it will default to "${Month}_${DoM},_${Year}_${Time}"

For example, suppose the following lines are in the "areas" script:

```
GifList="chlor_a_SeaDAS K_490_SeaDAS"
GifLabelOpts="20 30 3"
chlor_a_SeaDAS_GifLabelOpts="40 32 4"
```

In this example, the user has requested that the chlor_a_SeaDAS and K_490_SeaDAS browse images be generated. Each will have the default label written to it. The chlor_a_SeaDAS product will have the label written at (20,30) using color index 3 while the K_490_SeaDAS product will have the label written at (40,32) using color index 4.

${prod}_GifColorTable

The colortable used can be set by defining this variable. If not set, then it will default to $AutoData/color/ct_$prod.ct. If that file is not found or is unreadable, then it will default to a linear colortable.

ImgSamples and ImgLines

These are used to override the default sizes of the browse images. By default, these values will be calculated such that the browse image will be a integral zoom factor smaller than the original image that will be smaller than 640 by 640. For example, if the map size is 600 by 300, then the browse image will be 600 by 300. If the map size is 2430 by 1810, then the browse image will be 640 by 476. If the map size is 1500 by 1500, then the browse image will be 640 by 640.

Program Variables

These variables define the programs used by swfScripts. The user can overide these to test a new version of a program. They are defined in swfInit.

ApsInfo

Set to the name of the satellite specific program used to obtain information from input Level-1 file. Defaults to $AutoBin/swfInfo.

MSl12 Set to the name of the program used to do the atmospheric correction and generate the remote sensing reflectance products. Defaults to $AutoBin/swfCase2.

MosInfo

Set to the name of the program used to obtain information from MOS file. Defaults to $AutoBin/swfInfo.

CONFORMANCE

These script functions attempt to conform to the IEEE 1003.2 POSIX Shell Standard. They were, however, developed and tested using the Bourne Shell and Korn Shells under IRIX 5.3 and IRIX 6.5 respectively.

SEE ALSO

apsScripts(1) **daylight**(1), **filefmt**(1), **hdf**(1), **maps**(1), **nsLabel**(1), **nsOverlay**(1), **nsPicToGIF**(1), **imgMap**(1), **imgReformat**(1), **imgBrowse**(1), **imgSDStoImg**(1), **mosInfo**(1), **mosCase2**(1),

NAME

swfArea − determine file extents of geographical area

SYNOPSIS

swfArea [-M mapFile] mapName inFile

DESCRIPTION

Determine the file extents (start/stop pixel/line) of a SeaWiFS file (still in sensor projection, i.e. L1A, L2, etc.) that covers a map.

SwfArea begins by reading in the map from the mapFile. If the file can not be opened or the named map is not in the file, a diagnostic is printed and the program will exit.

Next, the SeaWiFS file is opened and the navigation information initialized. If unable to open the SeaWiFS file or retrieve the navigation information from it, the program will print a diagnostic and exit. The navigation to be read includes the SDS arrays "orb_vec", "scan_ell", "sen_mat", and "tilt".

Once the navigation has been set, **swfArea** reads in every 64th scan line, and using every 64th sample, determines if that point falls within the desired map. From this, the smallest box (modulo 64) that will cover the box will be determined. These file extents will be printed to the screen.

If the 64-sided box fails or the user has selected a refined coverage, **swfArea** will rescan the entire image (if 64-sides failed) or the box determined previously (if user selected refined coverage) using a small 5-sided box. If the file extents are found they are printed or the message "No coverage." If the file extents are the original input file, then the message will be "Complete coverage."

A third pass, which may be quite computer intensive, uses a reverse mapping to determine the file extents. It scans through the entire map image to determine where that pixel lies in the SeaWiFS file. For large map areas this computation can require large resources (i.e. memory and CPU time). The user can select this pass directly by using the exact option (-e) or allow it to be used after the first two passes have failed (-3).

OPTIONS

-e Do a reverse mapping from the map to the SeaWiFS file to determine its file extents. May be more computer intensive depending on the selected map.

-l Don't output start/stop line locations

-M mapFile
Use the given map file to find mapName. Defaults to $AUTO_DATA/maps.hdf

-p Don't output start/stop pixel locations

-r Refine search to within plus or minus 5 samples/lines.

-v Make output verbose.

-3 If passes one and two fail, use pass 3. This is mainly useful for maps that are smaller than the 5-sided box. (Why use SeaWiFS, then?)

--version

Print out version and exit.

ENVIRONMENTAL VARIABLES

AUTO_DATA

The location of the APS data directory.

EXAMPLES

```
$ swfArea -M maps.hdf MissBight S2000144175835.L1A_HNAV
257 835 1793 2177
$ swfArea -M maps.hdf EastSea S2000144175835.L1A_HNAV
No coverage
$ export AUTO_DATA=/usr/local/aps/data
$ swfArea MissBight S2000144175835.L1A_HNAV
257 835 1793 2177
$ swfArea -r MissBight S2000144175835.L1A_HNAV
301 783 1849 2177
```

SEE ALSO

MSl12(1)

NAME

swfArnone − creates images of inherent optical properities

SYNOPSIS

swfArnone [options] inFile [ouFile]

DESCRIPTION

The swfArnone program will generate inherent optical properties using the Arnone algorithm. The input file must be an HDF file with the SDSs, "rrs_412", "rrs_443", "rrs_490", "rrs_510", "rrs_555", "rrs_670", and "rrs_765", which represent the remote sensing reflectances at 412, ..., 765 nm. By default, the input remote sensing reflectances are used to estimate the chlorophyll concentration using the SeaBAM (OC2 v2) algorithm, however, the user can request to use another input chlorophyll image using the −c option.

By default, twelve output products, six total absorption and six backscattering at the SeaWiFS wavelengths of 412, 443, 490, 510, 555, 670, are written back to the input file as SDS's using the following nomenclature ‘‘a_XXX_arnone’’ and ‘‘bb_XXX_arnone’’, where ‘‘XXX’’ is replaced by the above values. Optionally, the user may request the difference product of the total absorption at 443 minus the total absorption at 412, as well as, total scattering and beam attenuation products at all six wavelengths.

OPTIONS

-a A bit-field representing flags for selecting one of the 6 possible total absorption outputs. For example, "−a 5" will select only the a_412 and a_490 products.

-b A bit-field representing flags for selecting one of the 6 possible backscatter outputs. For example, "−b 16" will select only the bb_555 product.

-B A bit-field representing flags for selecting one of the 6 possible total scattering outputs. For example, "−B 16" will select only the b_555 product. The nomenclature for these SDS's are ‘‘b_XXX_arnone’’.

-C A bit-field representing flags for selecting one of the 6 possible beam attenuation outputs. For example, "−B 9" will select only the c_412 and c_510 product. The nomenclature for these SDS's are ‘‘c_XXX_arnone’’.

-c Name of input SDS containing chlorophyll-a data. This option is used to specify an input chlorophyll-a image. Otherwise, the input remote sensing reflectances are used with the SeaBAM chlorophyll algorithm to estimate the chlorophyll concentration.

-d Select the a_443 − a_412 difference image for output. This SDS is called "adiff_arnone".

-r The nomenclature for the eight input remote sensing reflectance SDSs. The characters XXX will be replaced by the three digits representing the wavelengths. For example, "−r rrs_XXX" will select remote sensing reflectance arrays: "rrs_412", "rrs_443", etc. **(currently unimplemented)**

-v Verbose

--version
Print out version and exit.

REFERENCES

The Arnone algorithm is unpublished.

EXAMPLES

The following example appends twelve products (a_XXX and bb_XXX) for the first six SeaWiFS wavelengths to the input file.

```
$ swfArnone rrs.hdf
```

In this example the output file, ‘‘test.hdf’’, the total absorption at 412 nm, total scattering at 555 nm and beam attenuation at 670 nm, are created.

```
$ swfArnone -a 1 -B 16 -C 32 rrs.hdf test.hdf
```

SEE ALSO

MSl12(1), **swfCarder**(1)

NAME

swfCarder − create images of inherent optical properities

SYNOPSIS

swfCarder [options] inFile [ouFile]

DESCRIPTION

The swfCarder program will generate inherent optical properties using the Carder algorithm. This algorithm calculates aph675 and ag400 algebraically from Rrs model equations. Chlorophyll−a is then calculated from aph675. The input file must be an HDF file with the SDSs, "rrs_412", "rrs_443", "rrs_490", "rrs_510", "rrs_555", and "rrs_670", which represent the remote sensing reflectances at 412, ..., 670 nm.

By default, twenty−five output products, six total absorption, six backscattering, six photoplankton absorption, and six detris/gelbstuff absorption at the SeaWiFS wavelengths of 412, 443, 490, 510, 555, 670 and a chlorophyll-a product, are written back to the input file as SDS's using the following nomenclature ‘‘a_XXX_carder’’, ‘‘bb_XXX_carder’’, ‘‘aph_XXX_carder’’, ‘‘adg_XXX_carder’’ and ‘‘chlor_a_carder’’ where ‘‘XXX’’ is replaced by the above values. Optionally, the user can request the total scattering and beam attenuation products at all six wavelengths.

OPTIONS

-a A bit-field representing flags for selecting one of the 6 possible total absorption outputs. For example, "-a 5" will select only the a_412 and a_490 products.

-b A bit-field representing flags for selecting one of the 6 possible backscatter outputs. For example, "-b 16" will select only the bb_555 product.

-B A bit-field representing flags for selecting one of the 6 possible total scattering outputs. For example, "−B 16" will select only the b_555 product. The nomenclature for these SDS's are ‘‘b_XXX_carder’’.

-c Flag to output the chlorophyll-a data. This SDS will be named ‘‘chlor_a_carder’’.

-C A bit-field representing flags for selecting one of the 6 possible beam attenuation outputs. For example, "−B 9" will select only the c_412 and c_510 product. The nomenclature for these SDS's are ‘‘c_XXX_carder’’

-g A bit-field representing flags for selecting one of the 6 possible detris and gelbustuf absorption outputs. For example, "-g 17" will select the adg_412 and adg_555 products.

-p A bit-field representing flags for selecting one of the 6 possible phytoplankton absorption outputs. For example, "-p 7" will select only the aph_412, aph_443, and aph_490 products.

-s Do iteration on the "s" term.

-v Turn on verbose output.

--version
Print out version and exit.

REFERENCES

Carder et al., *Reflectance model for quantifying chlorophyll−a in the presence of productivity degradation products*, JGR, 96(C11), 20599-20611, 1991.

Lee et al., *Model for the interpretation of hyperspectral remote sensing reflectance*, Appl. Opt, 33(24), 5721, 1994.

EXAMPLES

The following example appends twenty−five products (a_XXX_carder, aph_XXX_carder, adg_XXX_carder, bb_XXX_carder, and chlor_a_carder) for the first six SeaWiFS wavelengths to the input file.

```
$ swfCarder rrs.hdf
```

In this example the output file, ‘‘test.hdf’’, the total absorption at 412 nm, total scattering at 555 nm and beam attenuation at 670 nm, are created.

```
$ swfCarder −a 1 −g 0 −p 0 −B 16 −c −C 32 rrs.hdf test.hdf
```

SEE ALSO

MSl12(1), **swfArnone**(1)

NAME

swfGetElements − download elements.dat file from NASA/GSFC

SYNOPSIS

swfGetElements

DESCRIPTION

The script **swfGetElements** is used to automatically retreive the `elements.dat` file needed by the swfL1AGen program. It is called from with `swfScripts`. This script *requires* the GNU wget package.

OPTIONS

--version

Print out version and exit.

DIAGNOSITCS

If the `$AUTO_DATA/seawifs/swfmops` directory does not exist or lacks user write permissions, the script will fail. If the `wget` program fails, a copy of the download will be printed to stdout.

SEE ALSO

swfL1AGen(1), **swfScripts**(1)

NAME

swfInfo − query information about a SeaWiFS Level−1A file

SYNOPSIS

swfInfo [option] swfFile

DESCRIPTION

This program is used to dump information about a SeaWiFS data file. With no options the program will print out a series of parameters. A single parameter can be single with the option. The options are succint as they were designed with shell scripting in mind.

OPTIONS

-day Day of month of input file.

-doy Day of year of input file.

-hour 2-digit time (HH) of input file.

-min 2-digit time (MM) of input file.

-month 3-character month of input file. Months are "jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"

-name Generate a file name in the following format as swf.YYYY.MMDD.HHMM. This is a short cut version of using −sat, −year, −doy, and −time.

-sat 3-character satellite name. Names is "swf".

-time 6-digit time (HHMMSS) of input file.

-type Character code for datatype: "LAC", "GAC", "HRPT"

-year 2-digit year of input file.

--version
Print out version and exit.

EXAMPLES

```
$ swfInfo S1998291133955.L1A_GAC
Filename:        S1998291133955.L1A_GAC
Starting Time:   10/18/1998 13:39, 291
Ending Time:     10/18/1998 14:19, 291
Satellite:       swf
Datatype:        GAC
Total Scans:
$

$ swfInfo −year S1998291133955.L1A_GAC
1998
```

```
$
```

Here is how a Bourne shell script function might use **swfInfo** to set the name of the output files from the input file:

```
 set_name()
{
    sat=`swfInfo -sat $1`
    yr=`swfInfo -year $1`
    jday=`swfInfo -doy $1`
    time=`swfInfo -time $1`
    file=S$yr$jday$time.L1A_HNAV
}
```

NAME

swfMail − mail filter for transfer of Level−1A data to Goddard

SYNOPSIS

swfMail

DESCRIPTION

The swfMail program is used to filter the mail communication that occurs between the receiving station and Goddard.

NAME

swfScripts – standard SeaWiFS processinig scripts

DESCRIPTION

swfScripts provides Bourne shell scripting functions to process SeaWiFS data from a NASA SeaWiFS Level−1A file. The swfProcess script function is the main user interface and provides for the standard processing steps. These steps are:

1 Initialize variables, check for programs, etc
2 Verify input file is NASA SeaWiFS Level−1A file using filefmt program.
3 Determine if file covers user defined map using swfArea program.
4 Get time information from file using swfInfo program.
5 Create parameter file for MSl12
6 Optionally, add daily ancillary data
7 Run MSl12 program to atmospherically correct Level−1 data (Level−2)
8 Warp Level−2 file to Level−3 using imgMap
9 If user had defined `L3BrowseList`, generate browse images for selected products.
10 If user has defined `SwfPostProcess`, it is now called. (Normally, not defined.)
11 If user has define `TSProd`, generate time series
12 Store HDF product file in the `DATA_BASE` and the browse image in the `IMAG_BASE`.
13 Generate Level−4 data adding in this new Level−3 file.

SETUP

A minimalist executable script for processing SeaWiFS data must source both the apsScripts and swfScripts located in the APS bin directory, call the script function swfProcess passing it the name of the file name containing the SeaWiFS data and define at least two script variables. Many other variables can be optionally set to modify the normal mode of operations. These must be set prior to the call to swfProcess.

```
#!/bin/sh
test -z "$AUTO_DIR" && return 1
AUTO_BIN=${AUTO_BIN:=AUTO_DIR/bin}
MapName=ChesapeakeBay
swfProcess $1 $0
```

The script must be placed in the $AREAS_PROC directory which is normally the directory `areas` located in the main APS directory. The script must have execution permissions.

REQUIRED VARIABLES

These variables are required to process an area. They provide the script an area to process. Most of the remaining variables have defaults which can be overridden. They are described in the next section.

MapName
This is the name of the image map stored in the file $MapFile.

OPTIONAL VARIABLES

The following sections are variables that have defaults which the user can override to change the behaviour of the default processing. They are grouped together by subject.

Product Selection

This is probably the first thing that anyone will want to change. By default, the SeaWiFS list of products both Level−3 and Level−4 are defined in a UNIX text files. Similiar the default browse images are also defined. The user can override these list of products in several ways.

First the use can actually modify the data files directly, or use the variables `L3ProdFile`, `L3ProdList`, `L4ProdFile`, or`L4ProdList`. Any any case, the user can select from any of the products listed below. Also, if a product is listed as a Level−4 product, it must also be listed as a Level−3 product.

Arnone IOP Products

a_412_arnone	Total Absorbtion at 412nm using Arnone
a_443_arnone	Total Absorbtion at 443nm using Arnone
a_490_arnone	Total Absorbtion at 490nm using Arnone
a_510_arnone	Total Absorbtion at 510nm using Arnone
a_555_arnone	Total Absorbtion at 555nm using Arnone
a_670_arnone	Total Absorbtion at 670nm using Arnone
bb_412_arnone	Backscattering at 412nm using Arnone
bb_443_arnone	Backscattering at 443nm using Arnone
bb_490_arnone	Backscattering at 490nm using Arnone
bb_510_arnone	Backscattering at 510nm using Arnone
bb_555_arnone	Backscattering at 555nm using Arnone
bb_670_arnone	Backscattering at 670nm using Arnone
b_412_arnone	Total Scattering at 412nm using Arnone
b_443_arnone	Total Scattering at 443nm using Arnone
b_490_arnone	Total Scattering at 490nm using Arnone
b_510_arnone	Total Scattering at 510nm using Arnone
b_555_arnone	Total Scattering at 555nm using Arnone
b_670_arnone	Total Scattering at 670nm using Arnone
c_412_arnone	Beam Attenuation at 412nm using Arnone
c_443_arnone	Beam Attenuation at 443nm using Arnone
c_490_arnone	Beam Attenuation at 490nm using Arnone
c_510_arnone	Beam Attenuation at 510nm using Arnone
c_555_arnone	Beam Attenuation at 555nm using Arnone
c_670_arnone	Beam Attenuation at 670nm using Arnone

Carder IOP Products

a_412_carder	Total Absorbtion at 412nm using Carder
a_443_carder	Total Absorbtion at 443nm using Carder
a_490_carder	Total Absorbtion at 490nm using Carder
a_510_carder	Total Absorbtion at 510nm using Carder
a_555_carder	Total Absorbtion at 555nm using Carder
a_670_carder	Total Absorbtion at 670nm using Carder
bb_412_carder	Backscattering at 412nm using Carder
bb_443_carder	Backscattering at 443nm using Carder
bb_490_carder	Backscattering at 490nm using Carder
bb_510_carder	Backscattering at 510nm using Carder
bb_555_carder	Backscattering at 555nm using Carder
bb_670_carder	Backscattering at 670nm using Carder
b_412_carder	Total Scattering at 412nm using Carder
b_443_carder	Total Scattering at 443nm using Carder
b_490_carder	Total Scattering at 490nm using Carder
b_510_carder	Total Scattering at 510nm using Carder
b_555_carder	Total Scattering at 555nm using Carder
b_670_carder	Total Scattering at 670nm using Carder
c_412_carder	Beam Attenuation at 412nm using Carder
c_443_carder	Beam Attenuation at 443nm using Carder
c_490_carder	Beam Attenuation at 490nm using Carder
c_510_carder	Beam Attenuation at 510nm using Carder
c_555_carder	Beam Attenuation at 555nm using Carder
c_670_carder	Beam Attenuation at 670nm using Carder
aph_412_carder	Absorbtion due to phytoplankton at 412nm
aph_443_carder	Absorbtion due to phytoplankton at 443nm
aph_490_carder	Absorbtion due to phytoplankton at 490nm
aph_510_carder	Absorbtion due to phytoplankton at 510nm
aph_555_carder	Absorbtion due to phytoplankton at 555nm
aph_670_carder	Absorbtion due to phytoplankton at 670nm
adg_412_carder	Absorbtion due to detritis and gelbstuff at 412nm
adg_443_carder	Absorbtion due to detritis and gelbstuff at 443nm
adg_490_carder	Absorbtion due to detritis and gelbstuff at 490nm
adg_510_carder	Absorbtion due to detritis and gelbstuff at 510nm
adg_555_carder	Absorbtion due to detritis and gelbstuff at 555nm
adg_670_carder	Absorbtion due to detritis and gelbstuff at 670nm

QAA IOP Products

a_412_qaa	Total Absorbtion at 412nm using Carder
a_443_qaa	Total Absorbtion at 443nm using Carder
a_490_qaa	Total Absorbtion at 490nm using Carder
a_510_qaa	Total Absorbtion at 510nm using Carder
a_555_qaa	Total Absorbtion at 555nm using Carder
a_670_qaa	Total Absorbtion at 670nm using Carder
bb_412_qaa	Backscattering at 412nm using Carder
bb_443_qaa	Backscattering at 443nm using Carder
bb_490_qaa	Backscattering at 490nm using Carder
bb_510_qaa	Backscattering at 510nm using Carder
bb_555_qaa	Backscattering at 555nm using Carder
bb_670_qaa	Backscattering at 670nm using Carder
b_412_qaa	Total Scattering at 412nm using Carder
b_443_qaa	Total Scattering at 443nm using Carder
b_490_qaa	Total Scattering at 490nm using Carder
b_510_qaa	Total Scattering at 510nm using Carder
b_555_qaa	Total Scattering at 555nm using Carder
b_670_qaa	Total Scattering at 670nm using Carder
c_412_qaa	Beam Attenuation at 412nm using Carder
c_443_qaa	Beam Attenuation at 443nm using Carder
c_490_qaa	Beam Attenuation at 490nm using Carder
c_510_qaa	Beam Attenuation at 510nm using Carder
c_555_qaa	Beam Attenuation at 555nm using Carder
c_670_qaa	Beam Attenuation at 670nm using Carder
aph_412_qaa	Absorbtion due to phytoplankton at 412nm
aph_443_qaa	Absorbtion due to phytoplankton at 443nm
aph_490_qaa	Absorbtion due to phytoplankton at 490nm
aph_510_qaa	Absorbtion due to phytoplankton at 510nm
aph_555_qaa	Absorbtion due to phytoplankton at 555nm
aph_670_qaa	Absorbtion due to phytoplankton at 670nm
adg_412_qaa	Absorbtion due to detritis and gelbstuff at 412nm
adg_443_qaa	Absorbtion due to detritis and gelbstuff at 443nm
adg_490_qaa	Absorbtion due to detritis and gelbstuff at 490nm
adg_510_qaa	Absorbtion due to detritis and gelbstuff at 510nm
adg_555_qaa	Absorbtion due to detritis and gelbstuff at 555nm
adg_670_qaa	Absorbtion due to detritis and gelbstuff at 670nm

Bio-optical Products

K_490	Diffuse Attenuation at 490nm using 490/555
K_532	Diffuse Attenuation at 532nm using 490/555
chl_oc2	Chlorophyll−a concentration using OC2
chl_oc4	Chlorophyll−a concentration using OC4
chl_stumpf	Chlorophyll−a concentration using Stumpf
chl_carder	Chlorophyll-a concentration using Carder
chl_trees	Chlorophyll−a concentation using Trees
chl_octsc	chlorophyll−a concentration using the OCTS−C algorithm
chl_nn	chlorophyll−a concentration derived from pig_nn data
chl_ndpi	chlorophyll−a concentration derived from pig_ndpi data
pig_oc2	pigment concentration derived from chl_oc2
pig_oc4	pigment concentration derived from chl_oc4
pig_octsc	pigment concentration derived from chl_octsc
pig_nn	pigment concentration using neural network algorithm
pig_ndpi	pigment concentration using normalized diffence pigment index
par	photosynthetically active radiation
depth	water depth index
N_small_particles	number of small particles using Haltrin's algorithm
N_large_particles	number of small particles using Haltrin's algorithm
N_particles	number of small particles using Haltrin's algorithm
salinity	salinity using Arnone's algorithm
visibility	diver visibility using 1.6/c (McBride)

Land Products

ndvi	normalized difference vegetation index
evi	enhanced vegetation index
smoke	smoke index

Quality control Products

l2_flags	Flags
flags_carder	Flags for Carder algorithm

Atmospheric Products

aerindex	aerosol index
aer_model_min	minimum bounding aerosol model #
aer_model_max	maximum bounding aerosol model #
aer_model_ratio	model mixing ratio
aer_num_iter	number of aerosol iterations, NIR correction
glint_coeff	glint radiance normalized by solar irradiance
epsilon	retreived epsilon used for model selection
cloud_albedo	cloud albedo at 865 nm
taua_412	aerosol optical depth at 412
taua_443	aerosol optical depth at 443
taua_490	aerosol optical depth at 490
taua_510	aerosol optical depth at 510
taua_555	aerosol optical depth at 555
taua_670	aerosol optical depth at 670
taua_765	aerosol optical depth at 765
taua_865	aerosol optical depth at 865
angstrom_412	aerosol angstrom coefficents (412,865)
angstrom_443	aerosol angstrom coefficents (443,865)
angstrom_490	aerosol angstrom coefficents (490,865)
angstrom_510	aerosol angstrom coefficents (510,865)
angstrom_555	aerosol angstrom coefficents (555,865)
angstrom_670	aerosol angstrom coefficents (670,865)
angstrom_765	aerosol angstrom coefficents (765,865)
angstrom_865	aerosol angstrom coefficents (865,865)
eps_412	ratio of 412
eps_443	ratio of 443
eps_490	ratio of 490
eps_510	ratio of 510
eps_555	ratio of 555
eps_670	ratio of 670
eps_765	ratio of 765
eps_865	ratio of 865

Radiance Products

Transmittance Products

t_sol_412	Rayleigh-aerosol transmittance,sun to ground at 412
t_sol_443	Rayleigh-aerosol transmittance,sun to ground at 443
t_sol_490	Rayleigh-aerosol transmittance,sun to ground at 490
t_sol_510	Rayleigh-aerosol transmittance,sun to ground at 510
t_sol_555	Rayleigh-aerosol transmittance,sun to ground at 555
t_sol_670	Rayleigh-aerosol transmittance,sun to ground at 670
t_sol_765	Rayleigh-aerosol transmittance,sun to ground at 765
t_sol_865	Rayleigh-aerosol transmittance,sun to ground at 865
t_sen_412	Rayleigh-aerosol transmittance,ground to sensor at 412
t_sen_443	Rayleigh-aerosol transmittance,ground to sensor at 443
t_sen_490	Rayleigh-aerosol transmittance,ground to sensor at 490
t_sen_510	Rayleigh-aerosol transmittance,ground to sensor at 510
t_sen_555	Rayleigh-aerosol transmittance,ground to sensor at 555
t_sen_670	Rayleigh-aerosol transmittance,ground to sensor at 670
t_sen_765	Rayleigh-aerosol transmittance,ground to sensor at 765
t_sen_865	Rayleigh-aerosol transmittance,ground to sensor at 865
t_oz_sol_412	ozone transmittance,sun to ground at 412
t_oz_sol_443	ozone transmittance,sun to ground at 443
t_oz_sol_490	ozone transmittance,sun to ground at 490
t_oz_sol_510	ozone transmittance,sun to ground at 510
t_oz_sol_555	ozone transmittance,sun to ground at 555
t_oz_sol_670	ozone transmittance,sun to ground at 670
t_oz_sol_765	ozone transmittance,sun to ground at 765
t_oz_sol_865	ozone transmittance,sun to ground at 865
t_oz_sen_412	ozone transmittance,ground to sensor at 412
t_oz_sen_443	ozone transmittance,ground to sensor at 443
t_oz_sen_490	ozone transmittance,ground to sensor at 490
t_oz_sen_510	ozone transmittance,ground to sensor at 510
t_oz_sen_555	ozone transmittance,ground to sensor at 555
t_oz_sen_670	ozone transmittance,ground to sensor at 670
t_oz_sen_765	ozone transmittance,ground to sensor at 765
t_oz_sen_865	ozone transmittance,ground to sensor at 865
t_o2_412	total oxygen transmittance at 412
t_o2_443	total oxygen transmittance at 443
t_o2_490	total oxygen transmittance at 490
t_o2_510	total oxygen transmittance at 510
t_o2_555	total oxygen transmittance at 555
t_o2_670	total oxygen transmittance at 670
t_o2_765	total oxygen transmittance at 765
t_o2_865	total oxygen transmittance at 865

Reflectance Products

rhos_412	surface reflectance at 412
rhos_443	surface reflectance at 443
rhos_490	surface reflectance at 490
rhos_510	surface reflectance at 510
rhos_555	surface reflectance at 555
rhos_670	surface reflectance at 670
rhos_765	surface reflectance at 765
rhos_865	surface reflectance at 865
rrs_412	Remote Sensing Reflectance at 412
rrs_443	Remote Sensing Reflectance at 443
rrs_490	Remote Sensing Reflectance at 490
rrs_510	Remote Sensing Reflectance at 510
rrs_555	Remote Sensing Reflectance at 555
rrs_670	Remote Sensing Reflectance at 670
rrs_765	Remote Sensing Reflectance at 765
rrs_865	Remote Sensing Reflectance at 865
foq_412	f/Q correction to nadir at 412
foq_443	f/Q correction to nadir at 443
foq_490	f/Q correction to nadir at 490
foq_510	f/Q correction to nadir at 510
foq_555	f/Q correction to nadir at 555
foq_670	f/Q correction to nadir at 670
foq_765	f/Q correction to nadir at 765
foq_865	f/Q correction to nadir at 865

Ancilliary Data Products

windspeed	magnitude of wind at 10 meters
zwind	zonal wind speed at 10 meters
mwind	meridional wind speed at 10 meters
windangle	wind direction at 10 meters
water_vapor	precipital water concentration
humidity	relative humidity
pressure	barometric pressure
ozone	ozone concentration
fsol	solar distance correction (1-D, not an image)
solz	solar zenith angle
sola	solar azimuth angle
senz	satellite zenith angle
sena	satellite azimuth angle

L3ProdFile

This variable can be set to a UNIX text file that contains a list of products for this region of interest. See an existing `L3ProdFile` for example of format. The default value is `$AUTO_DATA/seawifs/seawifs_def_l2prod.dat"`

`L4ProdFile`

This variable can be set to a UNIX text file that contains a list of products for this region of interest. See an existing `L4ProdFile` for example of format. The default value is `$AUTO_DATA/seawifs/seawifs_def_l4prod.dat"`

`L3ProdList`

This is a space delimetered list of products to be written to the output data base. See the list above for possible product names.

For example, suppose we want to study the aerosol optical depth products to match up with sun photometer. Then we will add the following lines in our script (NOTE: these will be the *only* products in the output file – the default products will be ignored):

```
# Name all desired products

L3ProdList="taua_412 taua_443 taua_490 taua_510"
L3ProdList="$L3ProdList taua_555 taua_670 taua_765 taua_865"
```

L4ProdList

This is a space delimetered list of products to be written to the output Level−4 data bases. See the list above for possible product names. The products listed here should also be listed as an Level−3 product.

Projection

These variables control information about the projection for the Level−3 file and naming of the file based on projection.

MapFile

Name of file containing image map file. Defaults to `$AutoData/maps.hdf`.

MapExt

This is a string that is appended to the Level−3 file which is written to the database. Usually it is a three character extension all uppercase.

MinPixels, MinLines

Used to set the minimum pixels/lines that must be extracted from the SeaWiFS file by swfExtract to continue processing. These are used to insure that enough of the input file covers the area of interest. By default these are not defined and, therefore, no check is performed.

SwfAreaOpts

This allows the use of options to the **swfArea**(1) program to be added. However, this string sould not contain the -p, -l, or -M options.

Data Base

These variables control information about where the data base of Level−3 and Level−4 will reside and the structure of that data base.

Region This variable will be used to create the default data base directories. By default it set to `$MapName`.

SwfDataBase

This variable is used to indicate the location of the image data base for the generated product file. By default, it is set to:

`$DataBase/$Level/$Sensor/$Version/$Region/$Year/$Month`.

where,
`$DataBase` is defined in the `aps.conf` file and represents the top directory of the data base.
`$Level` is set to the string "lvl3" by swfInit.
`$Sensor` is set to the string "seawifs" by swfInit.
`$Version` is set to "2.4" by swfInit.
`$Region` is set to "$MapName" by swfInit.
`$Year` and `$Month` are set by `swfProcess` based on the input file.

The user can override `$SwfDataBase` since it is evaluated by the shell prior to use. For example, if the line:

```
SwfDataBase="\$DataBase/seawifs/\$Year"
```

is set in the areas script and we assume that `$DataBase` is set to `/data` and that for a particular file $Year has been set to 1999, then the product file will be moved to `/data/seawifs/1999`. Note that to use the variables, the user must ‘‘escape’’ the ‘$’ by inserting a ‘‘\’’.

SwfDAYDataBase

This variable is used to indicate the location of the Level−4 daily composites data base for the generated product file. By default, it is set to:

`$DataBase/$CompLevel/$Sensor/$Version/$Region/daily/$Year/$Month`.

where,
`$DataBase` is defined in the `aps.conf` file and represents the top directory of the data base.
`$CompLevel` is set to the string "lvl4" by swfInit.
`$Sensor` is set to the string "seawifs" by swfInit.
`$Version` is set to "2.4" by swfInit.
`$Region` is set to "$MapName" by swfInit.
`$Year` and `$Month` are set by `swfProcess` based on the input file.

The user may override `$SwfDAYDataBase` since it is evaluated by the shell prior to use.

SwfNDDataBase

This variable is used to indicate the location of the Level−4 weekly (8-day) composites data base for the generated product file. By default, it is set to:

`$DataBase/$CompLevel/$Sensor/$Version/$Region/weekly/$Year`.

where,
`$DataBase` is defined in the `aps.conf` file and represents the top directory of the data base.
`$CompLevel` is set to the string "lvl4" by swfInit.
`$Sensor` is set to the string "seawifs" by swfInit.
`$Version` is set to "2.4" by swfInit.
`$Region` is set to "$MapName" by swfInit.
`$Year` is set by `swfProcess` based on the input file.

The user may override `$SwfNDDataBase` since it is evaluated by the shell prior to use.

SwfMODataBase

This variable is used to indicate the location of the Level−4 monthly composites data base for the generated product file. By default, it is set to:

`$DataBase/$CompLevel/$Sensor/$Version/$Region/monthly/$Year`.

where,
`$DataBase` is defined in the `aps.conf` file and represents the top directory of the data base.
`$CompLevel` is set to the string "lvl4" by swfInit.
`$Sensor` is set to the string "seawifs" by swfInit.
`$Version` is set to "2.4" by swfInit.
`$Region` is set to "$MapName" by swfInit.
`$Year` is set by `swfProcess` based on the input file.

The user may override `$SwfMODataBase` since it is evaluated by the shell prior to use.

SwfYRDataBase

This variable is used to indicate the location of the Level−4 yearly composites data base for the generated product file. By default, it is set to:

`$DataBase/$CompLevel/$Sensor/$Version/$Region/yearly`.

where,
`$DataBase` is defined in the `aps.conf` file and represents the top directory of the data base.
`$CompLevel` is set to the string "lvl4" by swfInit.
`$Sensor` is set to the string "seawifs" by swfInit.
`$Version` is set to "2.4" by swfInit.
`$Region` is set to "$MapName" by swfInit.

The user may override `$SwfYRDataBase` since it is evaluated by the shell prior to use.

CmpOpt

This can be defined by the user to select the type of compression program to call for the output product file before it is moved to `$SwfDataBase. This option can be set to: "gzip", "compress", "bzip2" or "none". Only set CmpOpt` to a compression type that is available on user's machine. *Note:* If the user has defined `XNumChunks` or that variable is defined by `swfInit`, then the HDF file will *automatically* be internally compressed and this variable should not be used.

XNumChunks, YNumChunks

These are used to define the number of “chunks” in each direction which will be created by `imgReformat` program for the product files. By default, a large level−3 file will be rewritten in chunks (or “tiles”) with each chunk being compressed. A chunk will be no larger than 640 by 640. So, if the map image is 2430 by 1810, then imgReformat will create a total of 12 chunks (4 across and 3 down). A 1500 by 1500 map image will be divided into 9 chunks (3 across and 3 down). A 600 by 300 map image will *not* be chunked.

L3BrowseDir

This variable is used to indicate the location of for the browse images. By default, it is set to:
`$ImagBase/$Level/$Sensor/$Version/$Region/$Year/$Month`.
where,
`ImagBase` is set in the `aps.conf` file and represents the top directory of the browse data base.
`Level` is set to the string "lvl3" by swfInit.
`Sensor` is set to the string "seawifs" by swfInit.
`Version` is set to "2.4" by swfInit.
`Region` is set to "$MapName" by swfInit.
`Year` and `Month` are set by swfProcess based on the input file.

The user can override `L3BrowseDir` since it is evaluated by the shell prior to use. For example, if the line:

```
L3BrowseDir="\$ImagBase/browse/\$Year"
```

is set in the areas script and we assume that `ImagBase` is set to `/data` and that for a particular file `Year` has been set to 1999, then the browse image will be moved to `/data/browse/1999`.

Debugging

The following variables can be set to help the user debug problems or retain files that are normally removed.

swfVerbose

This variable can be set to control the amount of debugging output for each swfScript function. The value should be set between 0 and 9, with a value of nine calling the Bourne Shell command `set -x`.

KeepLogFile

This is the file that will accept all logging information. By default is it set to the input file (SeaWiFS Level-1A file) plus `.log`.

KeepLogFile

Define this variable to prevent `swfProcess` from removing the log file.

KeepParam

Define this variable to prevent `swfProcess` from removing any of the parameter files.

KeepL2

Define this variable to prevent `swfProcess` from removing the Level−2 file.

SeaWiFS Compositing Options

SwfDAYCompOpts

This allows the user to add specific options from the **imgMean**(1) program for the daily composites. The user should not use the following options, howver: -o, -v, -L, -c, -F, -H, -T, -m.

SwfNDCompOpts

Like SwfDAYCompOpts except for weekly composites.

SwfMOCompOpts

Like SwfDAYCompOpts except for monthly composites.

SwfYRCompOpts

Like SwfDAYCompOpts except for yearly composites.

SwfCompOpts

Like SwfDAYCompOpts except applied to all composites.

CompTypes

A string containing a list of desired composites. May contain any of the following strings (“daily” “weekly” “monthly”, “yearly”, or “none”). By default it is defined as:

CompTypes="daily weekly monthly yearly". If you don't want a specific composite you can remove it. Note, however, that there is some dependencies.

NDay Used to define the number of days in a weekly composite. Currently set to 8.

SeaWiFS Processing Parameters

CalFile Name of calibration file to be used during Level−2 processing. Defaults to $AUTO_DATA/seawifs/SEAWIFS_SENSOR_CAL.TBL-199909-time_dep_7jun99.

LandMask
Name of landmask file to be used during Level−2 processing. Defaults to $AutoData/seawifs/landmask.dat

WaterMask
Name of watermask file to be used during Level−2 processing. Defaults to $AutoData/seawifs/watermask.dat

SwfAtmOpts
This controls the atmospheric algorithm used.

ms78	Multi-scattering with 765/865 model selection
ms68	Multi-scattering with 670/865 model selection
ms78siegel	Multi-scattering with 765/865 model selection and Siegel NIR iteration
ms68siegel	Multi-scattering with 670/865 model selection and Siegel NIR iteration
ms78arnone0	Multi-scattering with 765/865 model selection and Arnone NIR iteration
ms78arnone1	Multi-scattering with 765/865 model selection and Arnone NIR (+aph) iteration
ms78arnone2	Multi-scattering with 765/865 model selection and Arnone NIR (+adg) iteration
ms78arnone	Multi-scattering with 765/865 model selection and Arnone NIR (+aph+adg) iteration
ms78arnone0412it	Multi-scattering with 765/865 model selection and Arnone NIR and Stumpf 412 iteration
ms78arnone1412it	Multi-scattering with 765/865 model selection and Arnone NIR (+aph) and Stumpf 412 iteratic
ms78arnone2412it	Multi-scattering with 765/865 model selection and Arnone NIR (+adg) and Stumpf 412 iteratic
ms78arnone412it	Multi-scattering with 765/865 model selection and Arnone NIR (+aph+adg) and Stumpf 412 it
ms78stumpf	Multi-scattering with 765/865 model selection and STUMPF NIR (+aph+adg)
ms78stumpf412it	Multi-scattering with 765/865 model selection and STUMPF NIR (+aph+adg) and STUMPF 4
ms78mumm	Multi-scattering with 765/865 model selection and MUMM NIR calculation

SwfParamFile
Name of file containing additional options to the MSl12 program. These should *not* contain the following options: par=, ifile=, ofile1=, l2prod1=, ofmt=, calfile= (use CalFile instead), land= (use LandMask instead), water= (use WaterMask instead), met1=, met2=, met3= (use SwfMetDir, SwfMetFile instead), ozone1=, ozone2=, ozone3= (use SwfOzoneDir, SwfOzoneFile instead). All others should present no problems. The user's options located in the file pointed to by SwfParamFile are appended the to parameter file automatically generated by swfScripts. See **MSl12**(1) for other possible options.

SwfOzoneDir, SwfOzoneFile
Normally, these are undefined and will cause the climatology ozone file located in the data/seawifs directory to be used.

To use the NRT ozone data, the user must download it from the Goddard DAAC. The variable SwfOzoneDir points to the location of the NRT ozone file(s). If SwfOzoneFile is not set, then swfProcess will use the input file's date to build the appropriate file name. If this file had been compressed using gzip or compress, it is uncompressed. If SwfOzoneFile is set, it must be an uncompressed file -- swfProcess sends it straight to l2gen, which will die if the file is incorrect.

If `SwfOzoneDir` is undefined, or the file `$SwfOzoneDir/$SwfOzoneFile` is not a regular file, then climatology file will be used.

SwfMetDir, SwfMetFile

Like `SwfOzoneDir` and `SwfOzoneFile` defined above but for the MET data.

Browse Image Variables

noBrowse

This keyword will force no browse images to be created. It must be set to the string "no", as in `noBrowe="no"`.

L3BrowseFile

This keyword selects the input ascii file for the default Level−3 browse image files. It defaults to `$AUTO_DATA/seawifs/seawifs_def_l2browse.dat`.

L4BrowseFile

This keyword selects the input ascii file for the default Level−4 browse image files. It defaults to `$AUTO_DATA/seawifs/seawifs_def_l4browse.dat`.

L3BrowseList

This is a list of whitespace delimetered Level−3 products which are converted to browse images. The products in this list *must* also be present in the `L3ProdList` variable. By default, the browse images listed in `L3BrowseFile` are created.

L4BrowseList

This is a list of whitespace delimetered Level−3 products which are converted to browse images. The products in this list *must* also be present in the `L3ProdList` variable. By default, the browse images listed in `L3BrowseFile` are created.

BrowseOpts

It defines the user's options for all browse image. For example, the user may remove the grids by adding the line:

```
BrowseOpts="-g"
```

to the scripts.

${prod}_BrowseOpts

This defines the user's options for a particular product. For example, to produce an chlorophyll−a browse image that is 300 pixels by 400 lines and has a data range from 0.01 to 10.0 using a log10 scale, add the line:

```
chl_oc4_BrowseOpts="-f log10 -r 0.01,10.0 -R 20,199 -s 300,400"
```

in the areas script. See **imgBrowse**(1) for more information.

Note: For the other products (those list in `$L3ProdList` above but not here), the user must provide the corresponding variables as there are no defaults.

Program Variables

These variables define the programs used by `swfScripts`. The user can overide these to test a new version of a program. They are defined in swfInit.

ApsInfo
: Set to the name of the satellite specific program used to obtain information from input Level-1 file. Defaults to `$AutoBin/swfInfo`.

SwfArea
: Set to the name of the program used to determine if a SeaWiFS file covers a map. Defaults to `$AutoBin/swfArea.`

`SwfL1L2`
: Set to the name of the program used to do the atmospheric correction and generate the remote sensing reflectance products. Defaults to `$AutoBin/MSl12`.

SwfInfo
: Set to the name of the program used to obtain information from SeaWiFS file. Defaults to `$AutoBin/swfInfo`.

CONFORMANCE

These script functions attempt to conform to the IEEE 1003.2 POSIX Shell Standard. They were, however, developed and tested using the Bourne Shell and Korn Shells under IRIX 5.3 and IRIX 6.5 respectively.

EXAMPLES

These script functions attempt to conform to the IEEE 1003.2 POSIX Shell Standard. They were, however, developed and tested using the Bourne Shell and Korn Shells under IRIX 5.3 and IRIX 6.5 respectively.

SEE ALSO

apsScripts(1) **daylight**(1), **filefmt**(1), **hdf**(1), **maps**(1), **imgMap**(1), **imgReformat**(1), **imgBrowse**(1), **MSl12**(1), **swfArea**(1), **swfInfo**(1),

NAME

swfSeadas − create bio-optical images from remote sensing reflectance

SYNOPSIS

swfSeadas [options] inFile ouFile

DESCRIPTION

The swfSeadas program will create bio-optical products of chlorophyll-a concentration and diffuse attenuation. By default, it creates the chlorophyll-a concentration using the SeaBAM oc2 algorithm, the diffuse attenuation at 490 nm using the SeaDAS 443/555 algorithm, and the diffuse attenuation at 532 nm using Mueller's 490/555 algorithm.

OPTIONS

--version

Print out version and exit.

NAME

swfStumpf − correct rrs spectrum

SYNOPSIS

swfStumpf [options] inFile ouFile

DESCRIPTION

The swfStumpf program will correct the Rrs spectrum using the rrs@412. It will also output Rick Stumpf's Southeast/Gulf of Mexico Case 2 chlorophyll-a product.

OPTIONS

-c Do not output Stumpf's chlorophyll-a product

-C Do not correct the remote sensing reflectance.

-d Output adg412 product

-p Output aph443 product

--version
Print out version and exit.

EXAMPLES

To correct the rrs spectrum in file1.hdf and write the results to file2.hdf, but do not output the chlorophyll-a product:

```
$ swfStumpf -c file1.hdf file2.hdf
```

To use the rrs in file1.hdf and plact the chlorophyll-a concentration product in the same file.

```
$ swfStumpf -C file1.hdf file2.hdf
```

SEE ALSO

MSl12(1)

NAME

swfVisibility − creates diver visibility image from beam attenuation

SYNOPSIS

swfVisibility [options] inFile [ouFile]

DESCRIPTION

The swfVisibility program will generate a diver visibility image using an algorithm from McBride. The input file must be an HDF file with ‘‘c_555_Carder’’ which represents the beam attenuation at 555 nm. (We have found that Carder’s beam attenuation product works better than the Arnone beam attenutation product). The output product will be called ‘‘visibility’’.

The McBride algorithm used here is a simple ratio algorithm.

OPTIONS

-v Verbose

--version
Print out version and exit.

REFERENCES

The McBride algorithm reference is unknown.

EXAMPLES

The following example appends the visibiilty product to the input file.

```
$ swfVisibility rrs.hdf
```

SEE ALSO

MSl12(1), **swfCarder**(1)